CLAIMS

I claim:

1). A method, comprising:

analyzing each routine, of a software program having a plurality of separately compilable routines, to create a plurality of local side-effect problems for each routine; and

merging the local side-effect problems to create a global side-effect problem.

2). The method of claim 1, further comprising:

computing a global solution to the global problem; and

splitting the global solution into local solutions.

3). The method of claim 2, further comprising:

determining for each routine, whether a pointer parameter within the routine is used to write to or read from a storage device.

4). The method of claim 3, further comprising:

determining for each routine whether the pointer parameter is used to derive a return value of the routine.

5). The method of claim 4, further comprising:

computing a lattice value associated with each of the pointer parameters, wherein the lattice value comprises one of a PURE effect; LOST effect; RETURN effect; OUT effect; IN effect; RETURN, OUT, and IN effect; RETURN and OUT effect; RETURN and IN effect; and OUT and IN effect.

6). The method of claim 5, further comprising:

providing the lattice values to an interprocedural analysis solver to optimize compilation of the software program.

7). The method of claim 6, further comprising:
representing the local side-effect problems as directed graphs having edges and vertices, wherein
each edge has an associated monotone transfer function;
each vertex has a vertex value, wherein the vertex value is one of formal parameter, implicit parameter, local pointer variable, or gate parameter; and
a subset of vertices is marked with lattice values.

8). A computer-readable medium having stored thereon a plurality of instructions, said plurality of instructions when executed by a computer, cause said computer to perform:
analyzing each routine, of a software program having a plurality of separately compilable routines, to create a plurality of local side-effect problems for each routine; and
merging the local side-effect problems to create a global side-effect problem.

9). The computer-readable medium of claim 8 having stored thereon additional instructions, said additional instructions when executed by a computer, cause said computer to further perform:
computing a global side-effect solution to the global side-effect problem; and
splitting the global side-effect solution into local side-effect solutions.

10). The computer-readable medium of claim 9 having stored thereon additional instructions, said additional instructions when executed by a computer, cause said computer to further perform:

determining for each routine, whether a pointer parameter within the routine is used to write to or read from a storage device.

11). The computer-readable medium of claim 10 having stored thereon additional instructions, said additional instructions when executed by a computer, cause said computer to further perform:
determining for each routine whether the pointer parameter is used to derive a return value of the routine.

12). The computer-readable medium of claim 11 having stored thereon additional instructions, said additional instructions when executed by a computer, cause said computer to further perform,
computing a lattice value associated with each of the pointer parameters, wherein the lattice value comprises one of a PURE effect; LOST effect; RETURN effect; OUT effect; IN effect; RETURN, OUT, and IN effect; RETURN and OUT effect; RETURN and IN effect; and OUT and IN effect.

13). The computer-readable medium of claim 12 having stored thereon additional instructions, said additional instructions when executed by a computer, cause said computer to further perform:
providing the lattice values to an interprocedural analysis solver to optimize compilation of the software program.

14). The computer-readable medium of claim 13 having stored thereon additional instructions, said additional instructions when executed by a computer, cause said computer to further perform:
representing the local side-effect problems as directed graphs having edges and vertices, wherein
each edge has an associated monotone transfer function;

each vertex has a vertex value, wherein the vertex value is one of formal parameter, implicit parameter, local pointer variable, or gate parameter; and

a subset of vertices is marked with lattice values.

15). A system, comprising:

a processor;

memory connected to the processor storing instructions for interprocedural side-effect analysis executed by the processor;

storage connected to the processor that stores a software program having a plurality of separately compilable routines,

wherein the processor analyzes each routine, of the software program, to create a plurality of local side-effect problems for each routine; and merges the local side-effect problems to create a global side-effect problem.

16). The system of claim 15, wherein the processor computes a global solution to the global problem; and splits the global solution into local solutions.

17). The system of claim 16, wherein the processor determines for each routine, whether a pointer parameter within the routine is used to write to or read from the storage device.

18). The system of claim 17, wherein the processor determines for each routine whether the pointer parameter is used to derive a return value of the routine.

19). The system of claim 18, wherein the processor:

computes a lattice value associated with each of the pointer parameters, wherein the lattice value comprises one of a PURE effect; LOST effect; RETURN effect; OUT effect; IN effect; RETURN, OUT, and IN effect; RETURN and OUT effect; RETURN and IN effect; and OUT and IN effect.

20). The system of claim 19, wherein the processor:
provides the lattice values to an interprocedural analysis solver to optimize compilation of the software program.

21). The system of claim 20, wherein the processor:
represents the local side-effect problems as directed graphs having edges and vertices, wherein
each edge has an associated monotone transfer function;
each vertex has a vertex value, wherein the vertex value is one of formal parameter, implicit parameter, local pointer variable, or gate parameter; and
a subset of vertices is marked with lattice values.